

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application.

**Listing of Claims**

Claims 1-10 (canceled).

Claim 11 (currently amended): A computer-readable storage medium having stored thereon computer instructions that, when executed by a computer, cause the computer to:

provide a simulated checkpoint queue, the simulated checkpoint queue being associated with a simulated mean time to recovery (MTTR) setting, the simulated checkpoint queue being an ordered list of one or more elements, each of the one or more elements representing a respective buffer, the ordered list having a head and a tail;

in response to a change to a first buffer, check if the first buffer is represented in the simulated checkpoint queue, and if the first buffer is not represented in the simulated checkpoint, queue, link an element that represents the first buffer to the tail of the simulated checkpoint queue.

Claim 12 (original): The computer-readable storage medium of Claim 11 further storing computer instructions that, when executed by a computer, cause the computer to:

provide a simulated write counter, the simulated write counter being associated with the simulated MTTR setting; and

determine if linking the element to the tail of the simulated checkpoint queue causes the simulated checkpoint queue to exceed a predetermined length; and

in response to determining that the simulated, checkpoint queue exceeds the predetermined length, remove an element from the head of the simulated checkpoint queue and increment the simulated write counter.

Claim 13 (original): The computer-readable storage medium of Claim 12, wherein the predetermined length being a dirty buffer limit.

Claim 14 (currently amended): The computer-readable storage medium of Claim 11 further storing computer instructions that, when executed by a computer, cause the computer to, in response to a write out of a ~~second~~ an additional buffer from volatile memory and storing in nonvolatile memory, check if the ~~second~~ additional buffer is represented in the simulated checkpoint queue, and if the ~~second~~ additional buffer is represented in the simulated checkpoint queue, remove the element representing the second buffer from the simulated checkpoint queue and increment the simulated write counter.

Claim 15 (currently amended): The computer-readable storage medium of Claim 14, wherein the write out of ~~second~~ additional buffer being caused by an incremental checkpoint operation.

Claim 16 (original): The computer-readable storage medium of Claim 10, wherein each element in the simulated checkpoint queue comprises:

a first identifier that identifies an associated buffer; and

a second identifier that identifies a journal entry in a redo log, the journal entry corresponding to the associated buffer.

Claim 17 (original): The computer-readable storage medium of Claim 16, wherein the elements in the simulated checkpoint queue are ordered according to each element's journal entry position in the redo log.

Claim 18 (original): The computer-readable storage medium of Claim 11 further storing computer instructions that, when executed by a computer, cause the computer to determine a dirty buffer limit for the simulated checkpoint queue, the dirty buffer limit specifying the length of the simulated checkpoint queue, the dirty buffer limit being determined from the simulated MTTR setting and historical operating data.

Claim 19 (original): The computer-readable storage medium of Claim 18, wherein the historical operating data comprises an average time to read one journal entry in a redo log.

Claim 20 (original): The computer-readable storage medium of Claim 18, wherein the historical operating data comprises an average time to read one buffer from nonvolatile memory to volatile memory.

Claim 21 (original): A system comprising:

a memory;

one or more processors coupled to the memory;

a simulated MTTR setting maintained in the memory;

a simulated checkpoint queue maintained in the memory, the simulated checkpoint queue being associated with the simulated MTTR setting, the simulated checkpoint queue being an ordered list of one or more elements, each of the one or more elements representing a respective buffer, the ordered list having a head and a tail; and

a simulated write counter maintained in the memory, the simulated write counter being associated with the simulated MTTR setting, the simulated write counter providing a count of the number of times an element is removed from the simulated checkpoint queue, wherein the element is removed from the simulated checkpoint queue in response to a write out of a buffer from volatile memory and storing in nonvolatile memory.

Claim 22 (original): The system of Claim 21, wherein the element removed from the simulated checkpoint queue represents the buffer written out from volatile memory and stored in nonvolatile memory.

Claim 23 (original): The system of Claim 22, wherein the buffer is written out from volatile memory and stored in nonvolatile memory as a result of an incremental checkpoint operation involving the buffer.

Claim 24 (currently amended): The system of Claim 21, wherein the buffer is written out from volatile memory to nonvolatile memory as a result of ~~a second~~ an additional element being linked to the simulated checkpoint queue, wherein linking the ~~second~~ additional element causes the simulated checkpoint queue to exceed a predetermined length.

Claim 25 (currently amended): The system of Claim 24, wherein the ~~second~~ additional element does not represent the buffer written out from volatile memory and stored in nonvolatile memory.

Claim 26 (currently amended): The system of Claim 24, wherein the ~~second~~ additional element is linked to the simulated checkpoint queue in response to a modification to a ~~second~~ an additional buffer, the ~~second~~ additional buffer being represented by the ~~second~~ additional element.

Claim 27 (original): The system of Claim 24, wherein the predetermined length being a dirty buffer limit, the dirty buffer limit being determined from the simulated MTTR setting and historical operating data.

Claim 28 (original): The system of Claim 27, wherein the historical operating data comprises an average time to read one journal entry in a redo log.

Claim 29 (original): The system of Claim 27, wherein the historical operating data comprises an average time to read one buffer from nonvolatile memory to volatile memory.

Claim 30 (new): A method implemented in a computer that comprises a database system, to simulate effect of a recovery time required for the database system to recover from a database failure, the method comprising:

providing in said computer, in addition to a normal checkpoint queue used in the database system for normal operation, a simulated checkpoint queue;

wherein the normal checkpoint queue comprises a plurality of buffers;

wherein the simulated checkpoint queue comprises one or more elements, each element in the simulated checkpoint queue representing a respective buffer that is or was in the normal checkpoint queue;

wherein the simulated checkpoint queue is associated with a setting for recovery time whose effect on runtime performance of the database system is being simulated in said computer; and

in response to detecting a change to a buffer in said normal checkpoint queue due to actual database transactions occurring within the database system under normal operating conditions, checking if the buffer is represented in the simulated checkpoint queue, and if the buffer is not represented in the simulated checkpoint queue, linking an element that represents the buffer to the simulated checkpoint queue.

Claim 31 (new): The method of Claim 30 further comprising:

providing a simulated write counter, the simulated write counter being associated with the setting for recovery time;

wherein the simulated write counter provides a count of a number of times any element is removed from the simulated checkpoint queue in response to a write out of the respective buffer from volatile memory to nonvolatile memory.

Claim 32 (new): The method of Claim 30 further comprising:

determining if linking the element to the simulated checkpoint queue causes the simulated checkpoint queue to exceed a predetermined length; and

in response to determining that the simulated checkpoint queue exceeds the predetermined length, removing another element from the simulated checkpoint queue.

Claim 33 (new): The method of Claim 32 further comprising:

providing a simulated write counter, the simulated write counter being associated with the setting for recovery time; and

incrementing the simulated write counter if the simulated checkpoint queue exceeds the predetermined length.

Claim 34 (new): The method of Claim 32, wherein the predetermined length is a dirty buffer limit.

Claim 35 (new): The method of Claim 30 further comprising, in response to detecting a write out of any buffer, from volatile memory and storing in nonvolatile memory, checking if said any buffer is represented in the simulated checkpoint queue, and if said any buffer is represented in the simulated checkpoint queue, removing the element representing said any buffer from the simulated checkpoint queue and incrementing the simulated write counter.

Claim 36 (new): The method of Claim 35, wherein the write out of said any buffer is caused by an incremental checkpoint operation.

Claim 37 (new): The method of Claim 30, wherein each element in the simulated checkpoint queue comprises:

a first identifier that identifies the respective buffer in the normal checkpoint queue;  
and

a second identifier that identifies a journal entry in a redo log, the journal entry corresponding to the respective buffer.

Claim 38 (new): The method of Claim 30, wherein the elements in the simulated checkpoint queue are ordered according to each element's journal entry position in a redo log in the database system.

Claim 39 (new): The method of Claim 30 further comprising determining a dirty buffer limit for the simulated checkpoint queue, the dirty buffer limit specifying the length of the simulated checkpoint queue, the dirty buffer limit being determined from the setting of the recovery time, and historical operating data.